

# **PDelivery**

## **Final Report**

**By:** Asem Okby & Khaled Yassen

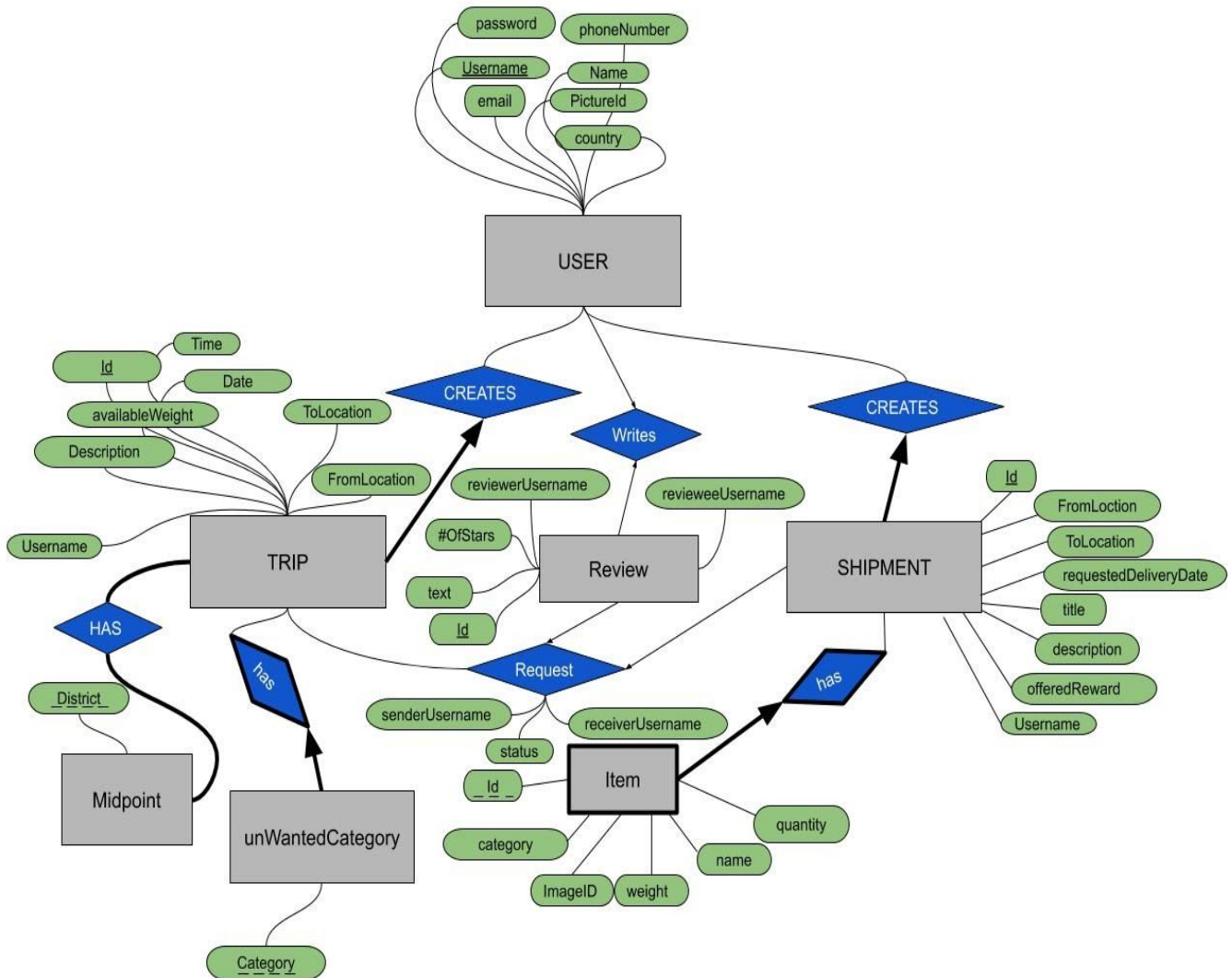
### **1) Motivation and Requirements**

- Our Project focuses on the issue of how cargo companies, such as DHL, XPO, and Ups, charge so much for a service that almost everyone can do. Another problem is that cargo companies are restricted by working hours which hinders a lot of people from getting what they need at the time they need.
- PDelivery is a tool that allows shoppers to connect with travelers. PDeli allows travelers to make a profit while allowing shoppers to not only save money but also get what they want! It also allows shoppers to put in the items they need, examples of the items, and also pictures of the items!
- **Requirements(Users will be able to..)**
  - A user could create a trip while specifying the categories he/she is not willing to deliver.
  - A user provides information about the package, its weight, the date, and the reward he is willing to give.
  - A user will be able to search for trips, and he will be presented with all the shipments that are available to deliver to that location.
  - A user will be able to search for shipments that need to be delivered to a specific location.
  - A user will be able to review another user when both complete a deal.
  - A user will be able to deliver more than one packet. Hence, maximizing his/her profit.
  - See suggested trips/shipments made based on his previous behaviors.

### **2) Conceptual Database Design**

- Each User has a username, name, password, phoneNumber, email, location, Birthday.
- Each Trip has an Id, from, to, date, time, availableWeight, description, and username.
- Each Shipment has an Id, from, to, requestedDeliveryDate, title, description, offeredReward, and username.
- Each Item has an id(partial key), a category, weight, a name, and a quantity.
- Each Review has an id, text, #OfStars, reviewerUsername, and revieweeUsername.
- Each Destination has a city/district.
- Each unWantedCategory has a category.
- Each User could create zero or more trips, but a Trip could be created by only one user.
- Each User could create zero or more shipments, A shipment could be created by only one user.
- Each User can review another user after a transaction(deal) is done.
- Each Trip has one(at least one) or more destinations.

- Each Trip has zero or more unwanted categories.
- Each Shipment has one(at least one) or more items.
- Each Item must belong to a shipment
- **FINAL VERSION OF ER:**



### 3) Logical Database Design

- **Initial Mapping of the ER Diagram into Relational Model**

Create table if not exists User(  
 Username VARCHAR(50),

```
    Name VARCHAR(50),
    Password VARCHAR(255) NOT NULL,
    Email VARCHAR(100) UNIQUE NOT NULL,
    Country VARCHAR(50),
    PhoneNumber VARCHAR(20),
    Birthday DATE,
    PRIMARY KEY (Username)
);
```

```
Create table if not exists Shipment(
    ID INTEGER AUTO_INCREMENT,
    Username VARCHAR(50),
    FromLocation VARCHAR(20) NOT NULL,
    ToLocation VARCHAR(20) NOT NULL,
    RequestedDeliveryDate Date NOT NULL,
    Title VARCHAR(100) NOT NULL,
    Description_ VARCHAR(250) NOT NULL,
    OfferedReward DECIMAL(10, 2) NOT NULL,
    PRIMARY KEY (ID),
    FOREIGN KEY (Username) REFERENCES User(Username) ON
DELETE CASCADE
);
```

```
Create table if not exists Item(
    ID INTEGER AUTO_INCREMENT,
    shipmentID INTEGER,
    Category VARCHAR(15) NOT NULL,
    ItemLink VARCHAR(300),
    Weight INTEGER NOT NULL,
    Name VARCHAR (50) NOT NULL,
    Quantity INTEGER NOT NULL,
    PRIMARY KEY (ID, shipmentID),
    FOREIGN KEY (shipmentID) REFERENCES Shipment(ID) ON DELETE
CASCADE
);
```

```
Create table if not exists Trip(
    Id INTEGER AUTO_INCREMENT,
    Username VARCHAR(50),
    FromLocation VARCHAR(20) NOT NULL,
    ToLocation VARCHAR(20) NOT NULL,
    Date DATE NOT NULL,
```

```

        Time Time NOT NULL,
        AvailableWeight INTEGER NOT NULL,
        Description VARCHAR(250) NOT NULL,
        unWantedCategories VARCHAR(300),
        PRIMARY KEY(Id),
        FOREIGN KEY (Username) REFERENCES User(Username) ON
DELETE CASCADE
);

```

```

Create table if not exists Midpoint(
    District CHAR(20) NOT NULL,
    TripID INTEGER AUTO_INCREMENT,
    PRIMARY KEY (District, TripID),
    FOREIGN KEY (TripID) REFERENCES Trip(ID) ON DELETE CASCADE
);

```

```

Create table if not exists Review(
    ID INTEGER AUTO_INCREMENT,
    TripID INTEGER,
    ShipmentID INTEGER,
    reviewerUsername VARCHAR(50), ##NEW
    revieweeUsername VARCHAR(50), ## NEW
    NumOfStars INTEGER NOT NULL,
    Text VARCHAR(500) NOT NULL,
    PRIMARY KEY (ID, TripID, ShipmentID),
    FOREIGN KEY (ShipmentID) REFERENCES Shipment(ID),
    FOREIGN KEY (TripID) REFERENCES TRIP(Id),
    FOREIGN KEY (reviewerUsername) REFERENCES USER(Username)
ON DELETE SET NULL,
    FOREIGN KEY (revieweeUsername) REFERENCES USER(Username) ON
DELETE SET NULL
);

```

```

Create table if not exists Request(
    ID INTEGER AUTO_INCREMENT,
    TripID INTEGER,
    ShipmentID INTEGER,
    senderRev INTEGER,
    receiverRev INTEGER,
    senderUsername VARCHAR(50),
    receiverUsername VARCHAR(50),
    status INTEGER,
    PRIMARY KEY (ID),

```

```

FOREIGN KEY (TripID) REFERENCES TRIP(Id),
FOREIGN KEY (ShipmentID) REFERENCES Shipment(ID),
FOREIGN KEY (senderUsername) REFERENCES USER(Username) ON
DELETE SET NULL,
FOREIGN KEY (receiverUsername) REFERENCES USER(Username)
ON DELETE SET NULL,
FOREIGN KEY (senderRev) REFERENCES Review(ID) ON DELETE
SET NULL,
FOREIGN KEY (receiverRev) REFERENCES Review(ID) ON DELETE
SET NULL
);

```

- **Functional Dependencies**

- 1) **User** table

- a)  $\{ID\} \rightarrow \{username\}$
- b)  $\{ID\} \rightarrow \{Email\}$
- c)  $\{ID\} \rightarrow \{phoneNumber\}$
- d)  $\{ID\} \rightarrow \{username, Email, phoneNumber\}$
- e)  $\{ID, username\} \rightarrow \{Email, phoneNumber\}$
- f)  $\{ID, Country\} \rightarrow \{phoneNumber\}$
- g)  $\{ID, Country\} \rightarrow \{username, Email, phoneNumber\}$

- 2) **Shipment** table

- a)  $\{ID\} \rightarrow \{Username, FromLocation, ToLocation, RequestedDeliveryDate, Title, Description, OfferedReward\}$
- b)  $\{ID\} \rightarrow \{FromLocation, ToLocation\}$
- c)  $\{ID\} \rightarrow \{RequestedDeliveryDate\}$
- d)  $\{ID\} \rightarrow \{Username\}$
- e)  $\{ID\} \rightarrow \{OfferedReward\}$
- f)  $\{ID\} \rightarrow \{Title, Description\}$
- g)  $\{ID, Username\} \rightarrow \{FromLocation, ToLocation, RequestedDeliveryDate, Title, Description, OfferedReward\}$

- 3) **Item** table

- a)  $\{ID\} \rightarrow \{category\}$
- b)  $\{ID\} \rightarrow \{category\}$
- c)  $\{ID\} \rightarrow \{category, weight\}$
- d)  $\{ID\} \rightarrow \{category, weight, Name\}$
- e)  $\{ID\} \rightarrow \{category, weight, Name, Quantity\}$
- f)  $\{ID, Name\} \rightarrow \{Category\}$
- g)  $\{ID, Name\} \rightarrow \{weight\}$

h)  $\{ID, Name, quantity\} \rightarrow \{weight\}$

5) **Trip** table

- a)  $\{ID, Username\} \rightarrow \{FromLocation, ToLocation, Date, Time, AvailableWeight, Description, Categories \text{ not wanted}\}$
- a)  $\{ID\} \rightarrow \{FromLocation, ToLocation\}$
- b)  $\{ID\} \rightarrow \{Date, Time\}$
- c)  $\{ID\} \rightarrow \{AvailableWeight\}$
- d)  $\{ID\} \rightarrow \{Description\}$
- e)  $\{ID\} \rightarrow \{Categories \text{ not wanted}\}$
- f)  $\{ID\} \rightarrow \{FromLocation, ToLocation, Date, Time, AvailableWeight, Description, Categories \text{ not wanted}\}$

6) **MidPoint** table:

- The relation has no functional dependencies since it is a **full-keyed** relation.

7) **Review** table:

- a)  $\{ID, tripID, shipmentID\} \rightarrow \{reviewUsername, revieweeUsername\}$
- b)  $\{ID, tripID, shipmentID\} \rightarrow \{text\}$
- c)  $\{ID, tripID, shipmentID\} \rightarrow \{text, NumOfStars\}$
- d)  $\{ID, TripID\} \rightarrow \{ShipmentID\}$
- e)  $\{ID, tripID, shipmentID\} \rightarrow \{reviewUsername, revieweeUsername, text, NumOfStars\}$
- f)

8) **Request** table:

- a)  $\{ID\} \rightarrow \{TripID, ShipmentID, senderRev, ReceiverRev, senderUsername, receiverUsername, Status\}$
- b)  $\{ID, TripID, Shipment\} \rightarrow \{SenderRev, ReceiverRev, senderUsername, receiverUsername, Status\}$

- **Normalization**

1) **User** table

- a) The table is in BCNF because there are no bad FDs. All the FDs contain the ID attribute on the left. Therefore, to prove that the table is in BCNF it is enough to show that the closure of the ID attribute contains all the attributes in the table. See the closure of the ID attribute below:
  - i)  $\{ID\}^+ = \{username, Email, phoneNumber\}$

2) **Shipment** table

- a) The table is in BCNF because there are no bad FDs. All FDs contain the ID attribute that its closure contains all attributes.

the closure of the ID attribute below:

- i)  $\{ID\}^+ = \{FromLocation, ToLocation, RequestedDeliveryDate, Title, Description, OfferedReward\}$

3) **Item** table

- a) The table is in BCNF since there are no bad FDs. The primary attribute's closure, that is ID, covers all of the table's attributes.

the closure of the ID attribute below:

- i)  $\{ID\}^+ = \{category, weight, Name, Quantity\}$

4) **Trip** table

- a) The table is in BCNF (for the same reasons as the tables above), however, it is not in **NF1**. The reason is the attribute's, 'CategoriesNotWanted', values are not atomic. We fix this problem by decomposing the table into two tables that are both in BCNF and NF1.

- i) **First table: Trip**(ID, FromLocation, ToLocation, Date, Time, AvailableWeight, Description)

- ii) **Second table: unwantedCategory**(TripID, category)

(1) This table is fully-keyed, and hence has no FDs at all.

- **To show that the decomposition is lossless and that it is dependency preserving**, we provide the following example:

- Below we show some tuples for each table:

- **unwantedCategory**

TripID	Category
1	Food
2	Elec.

- **Trip**

ID	Username	From Location	To Location	Date	Time	Available Weight	Description
1	Khaled	Egypt	Turkey	2020-08-11	12:10:00	15	I will travel, anything to deliver?
2	Asem	Egypt	Turkey	2020-08-11	12:10:00	15	I will deliver that for you

- When join is applied here, see the table below, all FDs still hold. So, the **decomposition is lossless**.
- And we added already tuples to each table, and performed join as shown below, and the FDs are not violated. So, the **decomposition is dependency preserving**.

ID	Username	From Location	To Location	Date	Time	Available Weight	Description	TripID	Category
1	Khaled	Egypt	Turkey	2020-08-11	12:10:00	15	I will travel, anything to deliver?	1	Elec.
2	Asem	Egypt	Turkey	2020-08-11	12:10:00	15	I will deliver that for you	2	Food

5) **MidPoint** table

- a) The table is in BCNF since it is a fully-keyed relation, meaning that there are no bad FDs for this table, and hence there are no bad FDs.

6) **Review** table

- a) This table is in BCNF as it also has an ID attribute that ensures there are no bad FDS. If its closure is produced, it would also cover all the attributes.
  - i) This table's closure:  
 $\{ID\}^+ = \{tripID, shipmentID, reviewUsername, revieweeUsername, text, NumOfStars\}$

7) **Request** table

- a) The table is in BCNF since all the FDs are good. All FDs contain the column ID on the left, and the closure is:
  - i)  $\{ID\}^+ = \{TripID, ShipmentID, senderRev, ReceiverRev, Sender, Receiver, Status\}$

## 4) Application Design and Implementation

- **Architecture**

- The architecture of this project is three-tier. The three layers are: presentation(application client), business logic(application server), and a



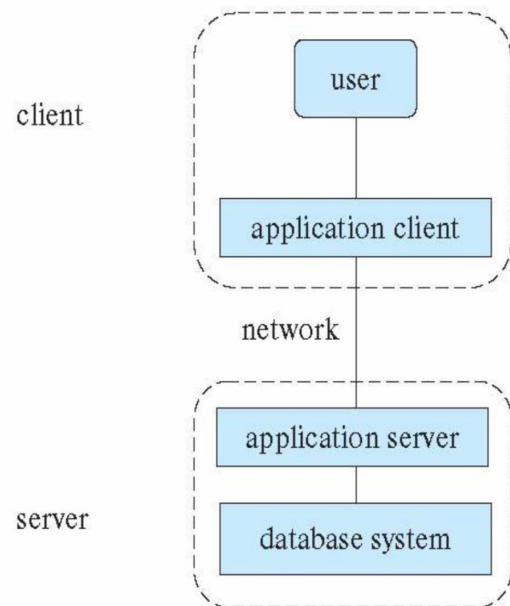
database layer(database system). The graph below shows clearly the layers.

- **Application**

**Client**(presentation layer): this layer is responsible for presenting the information about the services the application provides. It servers static files requested from the application server over the network.

- **Application server**(business logic layer): this layer is responsible for controlling the application's functionality. This layer handles the requests coming from the application client layer.

- **Database System**: this layer is responsible for storing the data. Here, we define our schemas so the application server could query any data while considering the schema of the tables.



- **SQL queries**

- **Validating a user** (condition here could be one of two columns: username, or email)

**select \***

**from User**

**Where condition = usernameOrEmail**

- Registering a user

- Check for validity of an email

**select \***

**from User**

**where Email= {email}**

- Check for the availability of a username

**select \***

**from User**  
**where Username= username**

- Inserting a user

**insert User(Name, Username, Email, Password,**  
**Country, PhoneNumber, Birthday, isActive)**  
**values(name, username, email, password, country,**  
**phoneNum, birthday, True)**

- Getting all information about a user

**select \***  
**from User**  
**where Username = username**

- Getting the trips of a user

**select \***  
**from Trip**  
**where Username = username**

- Getting the shipments of a user

**select \***  
**from Shipment**  
**where Username = username**

- Getting the reviews of a user

**select \***  
**from Review**  
**where revieweeUsername = username**

- Getting the requests of a user

**select \* from**  
**Request**  
**where senderUsername=username**  
**receiverUsername=username**

- Inserting a request

**insert into Request(TripID, ShipmentID, senderUsername,**  
**receiverUsername, status) values (tripId, shipmentId,**  
**'shipmentOwner, tripOwner, status)**

- Update a request on acceptance

**UPDATE Request**

**SET status = status**

**WHERE ID = reqId**

- Update a request on decline

**UPDATE Request**

**SET status = status + status**

**WHERE ID = reqId**

- Delete a request on Cancelation

**Delete**

**From Request**

**WHERE ID = reqId**

- Add a review

**insert into Review(tripID, ShipmentID, reviewerUsername,**

**revieweeUsername, NumOfStars, Text) values(TripID,**

**ShipmentID, reviewerUsername, revieweeUsername,**

**NumOfStars, Text)**

- Suggesting trips & shipments to users (the following is only for the trips, the queries of the shipments are similar)

**select FromLocation**

**from trip as t, User as u**

**where t.Username= username and u.Username = username**

**UNION**

**select ToLocation**

**from trip as t, User as u**

**where t.Username = username and u.Username = username**

**UNION**

**select FromLocation**

**from trip as t, User as u**

**where u.Username = username and FromLocation= u.Country**

**UNION**

**select ToLocation**

**from trip as t, User as u**

**where u.Username = username and ToLocation= u.Country**

- With the result of the above query(countries the user is interested in) we suggest trips for a user:

```
select *  
from trip  
where Username != username and (ToLocation =  
country or FromLocation = country
```

- Handling a search(the following is for the shipments, the trips' is similar)

```
select *  
from Shipment  
where Username != username and FromLocation =  
from and ToLocation = to and RequestedDeliveryDate  
<= date
```

- Updating Information of a user

```
UPDATE User  
set Name = name, Email = email, Country=location  
where Username = username
```

- Insert a trip

```
insert Trip(Username, FromLocation, ToLocation,  
Date, Time, AvailableWeight, Description) values(username,  
from, to, date, time, weight, description)
```

- Insert unWanted categories

```
insert unwantedcategory(tripID, Category)  
values(tripid, category)
```

- Insert Midpoints

```
insert midpoint(District, TripID) values(district  
id)
```

- Insert a shipment

```
insert shipment(Username, FromLocation, ToLocation,  
RequestedDeliveryDate, Title, Description_, OfferedReward)  
values(username, from, to, date, title, description, reward)
```

- Insert Items

```
insert item (shipmentID, Category, Weight,  
Name, Quantity, ItemLink) values (id, category,  
weight, name, quantity, itemlink)
```

- Get all midpoints of a trip

```
select District  
from Midpoint  
where TripID = id
```

- Get unwanted categories of a trip

```
select Category  
from UnWantedCategory  
where tripID = id
```

- Get items of a shipment

```
select *  
from Item  
where shipmentID = id
```

- **Programming Languages and DBMS**

- **Programming Languages:** Python(Flask framework) and Javascript(ReactJs framework)
- **DBMS:** MySQL

- **Data size and Source**

- No, we have not fetched any data.
- Yes, we have entered the data manually since there was no proper way for us to randomly fill the database with the data while maintaining the logic of them. We already had quite some data since we kept populating the database while testing the website's functionality.

- **Features**

- **SignUp:** Here we sign up a user, then log in the newly created user.

Welcome!

SignUp

Ahmed Omar

ahmedomar

ahmedomar@yahoo.com

....

Birthday

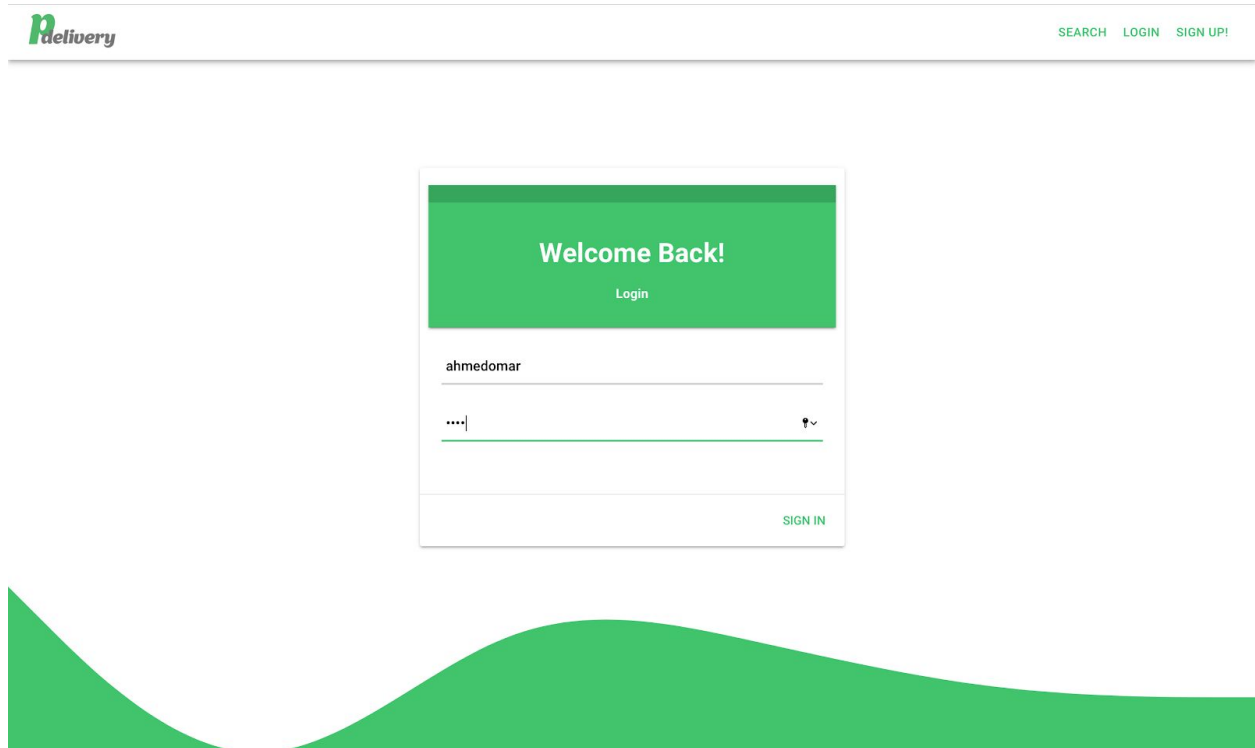
May 12th

Choose a country

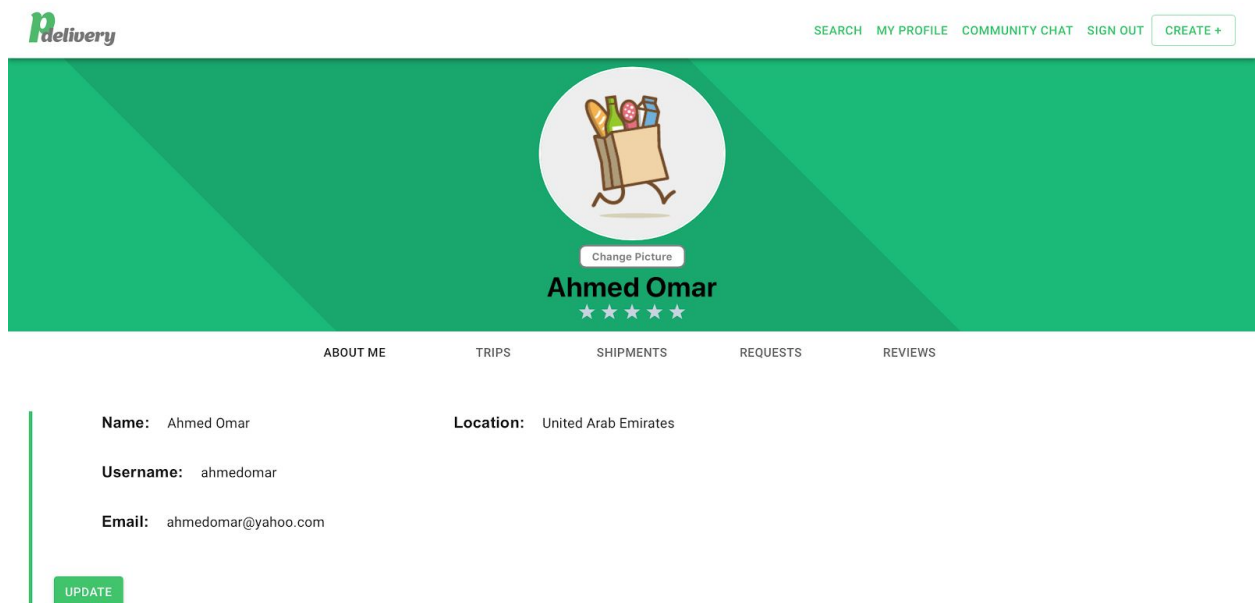
United Arab Emirates

+971 7 583 5793

SIGN UP



- The user's profile looks like the following:



- The user may update his information or his profile picture:



Change Picture

**Ahmed Asim**



ABOUT ME

TRIPS

SHIPMENTS

REQUESTS

REVIEWS

**Name:** Ahmed Asim

**Location:** United Arab Emirates

**Username:** ahmedomar

**Email:** ahmedomar@yahoo.com

UPDATE

- The user may create a Trip:

### Create Trip

Please provide the information of your trip.

Choose a country  
Germany

Choose a country  
Turkey

Departs At  
May 31st 03:49 PM

10 Kg

Available Weight  
Categories You Won't Carry

Food

Countries You Will Pass By

Albania


Description

I will be traveling soon to Turkey. I will also pass by Albania. Send a request if you are interested.

CANCEL ADD

- The user can then display his trips in his profile:





Change Picture

**Ahmed Asim**

★★★★★


ABOUT ME

TRIPS

SHIPMENTS

REQUESTS

REVIEWS



@ahmedomar

2020-05-31

Germany ✈

Turkey

Midpoints: Albania

Depart Date: 2020-05-31

Time: 12:49:26

Available Weight: 10 KG

Unwanted Categories: Food

I will be traveling soon to Turkey. I will also pass by Albania. Send a request if you are interested.

- The user may create a shipment:

Create Shipment

Choose a country

Turkey

Choose a country

Egypt

On

May 31st

Title

some documents

Offered Reward

\$ 10



Description

I need someone going to Egypt soon. I want to deliver some documents.

Shipment Items

Items

Q Search X +

Actions	Item Name	Item Link	Item Weight(KG)	Quantity(Piece)	Category
 	Quiz	https://lms.sehir.edu.tr/login/index.php	.5	2	Other

UPLOAD PICTURES

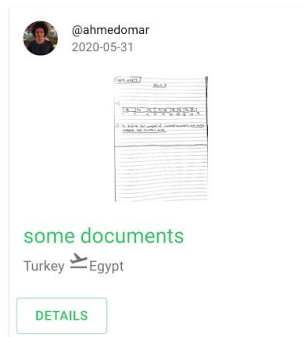
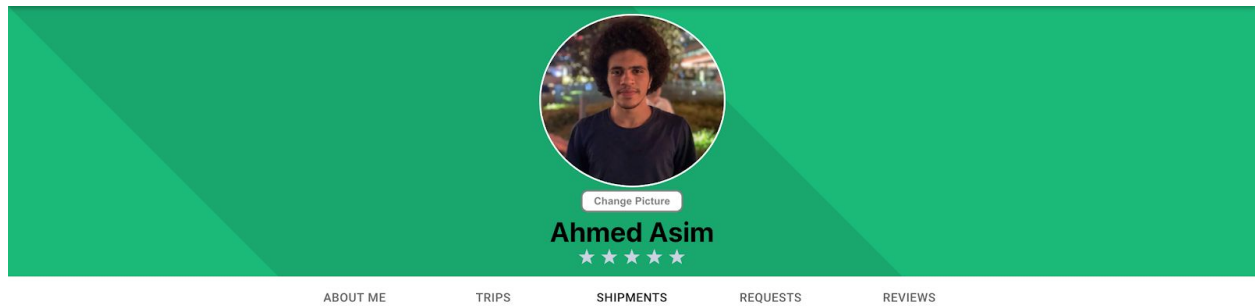
Asem\_Okby1.jpeg

DONE

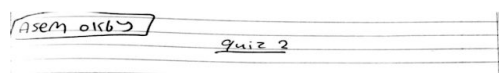
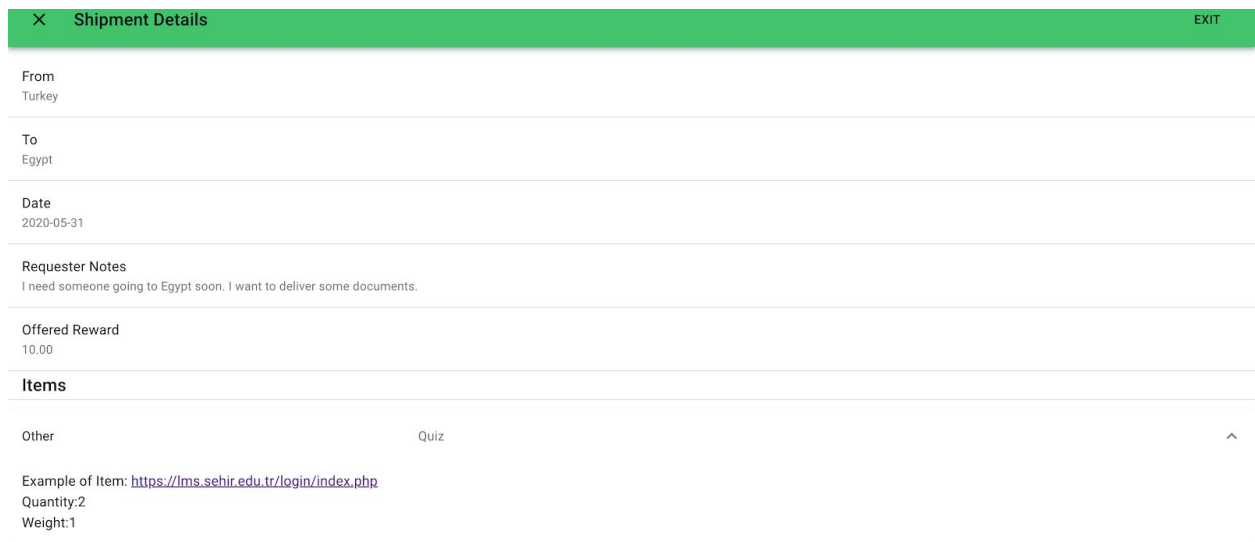
CANCEL

ADD


- Then view it on his profile as with the trip:



- also, a user could display the details of the shipment:



- A user may go on search for a trip or a shipment, but before this some suggested trips or shipments will be displayed:

SEARCH MY PROFILE COMMUNITY CHAT SIGN OUT CREATE +

# Where to Next?

Choose a country


Choose a country

Until  
May 30th

Trips

SEARCH

## SUGGESTED



@Beyza\_Yil  
2020-05-28

### Turkey ↗ Egypt

**Midpoints:**  
**Depart Date:**2020-05-28  
**Time:**18:19:49  
**Available Weight:**9 KG  
**Unwanted Categories:**Cameras

Who want to send something to Egypt? Send a request please.

SEND REQUEST




@Beyza\_Yil  
2020-05-28

### Turkey ↗ Sweden

**Midpoints:**Finland  
**Depart Date:**2020-05-28  
**Time:**18:19:49  
**Available Weight:**10 KG  
**Unwanted Categories:**Accessories,Kitchen Stuff

I will fly to Sweden soon from Turkey and pass by Finland. Send a request if you are interested.

SEND REQUEST




@Elifoo  
2020-05-31

### Turkey ↗ Egypt

**Midpoints:**  
**Depart Date:**2020-05-31  
**Time:**18:25:59  
**Available Weight:**3 KG  
**Unwanted Categories:**Kitchen Stuff,Stationary


I am heading to Egypt soon. Hit me up if you are interested.

SEND REQUEST




@Elifoo  
2020-05-31

### Turkey ↗



@EmreBiyo  
2020-05-31

### Turkey ↗ Egypt



@EmreBiyo  
2020-05-28

### Turkey ↗ Finland

- If the user wishes to search for a specific trip or shipment:

# Where to Next?

Choose a country  
Turkey


Choose a country  
Egypt

Until  
May 30th

Trips

SEARCH

## Results



@Beyza\_Yil  
2020-05-28

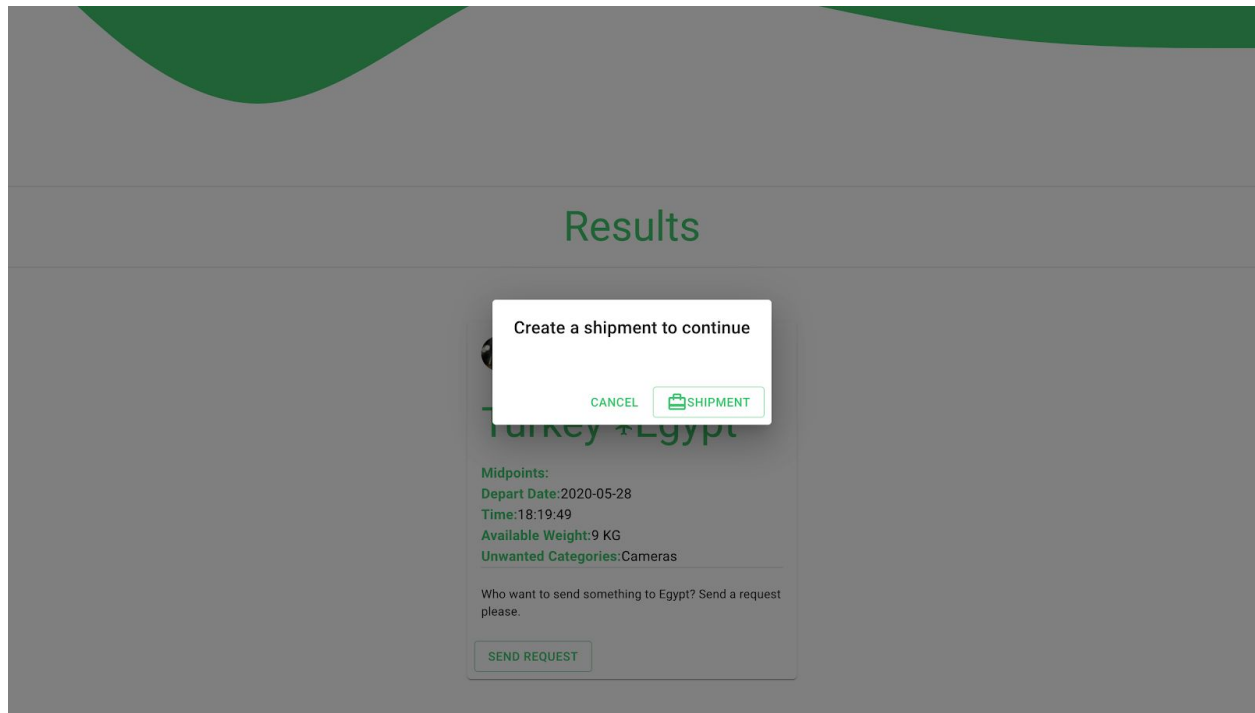
### Turkey ↗ Egypt

**Midpoints:**  
**Depart Date:**2020-05-28  
**Time:**18:19:49  
**Available Weight:**9 KG  
**Unwanted Categories:**Cameras

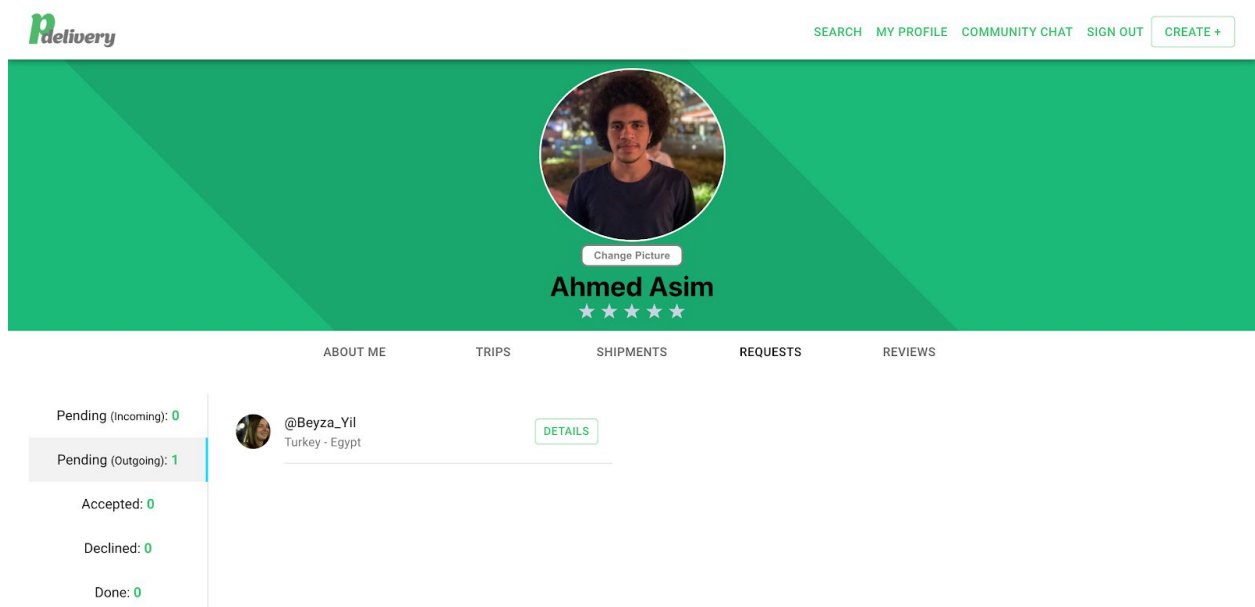
Who want to send something to Egypt? Send a request please.

SEND REQUEST

- Then, the user may send a request. Before this the user will be asked to create a shipment if it is a trip and create a trip otherwise:



- When done, the user will be able to view the requests in his/her profile.



- The other person will be able then to view the details of the request, and decline or accept.

× Details ACCEPT DECLINE

Turkey ↕ Egypt

### Trip

@Beyza\_Yil

Midpoints: Depart Date: 2020-05-28 Depart Time: 18:19:49 Available Weight: 9 KG Unwanted Categories: Cameras

Description: Who want to send something to Egypt? Send a request please.

### Shipment


@ahmedomar


Date: 2020-05-28 Reward: 10.00

Description: please accept my request.

Item 1: quiz Category: Other Weight: 1 KG Quantity: 1

- When the other person accepts the request, it is moved to the accepted requests tab.

 SEARCH MY PROFILE COMMUNITY CHAT SIGN OUT CREATE +

  
Change Picture  
**Beyza Yillmaz**  
★★★★★

ABOUT ME TRIPS SHIPMENTS REQUESTS REVIEWS


Pending (Incoming): 0

Pending (Outgoing): 0

Accepted: 1

Declined: 0

Done: 0



@ahmedomar  
Turkey - Egypt

DETAILS

- Now, if the user clicks on Details of an accepted request, the contact information of the other person is displayed.

×

Details

CANCEL

Turkey ↕ Egypt

Get in touch with ahmedomar

Email: ahmedomar@yahoo.com

Phone Number: +97175835793

Trip

@Beyza\_Yil

Midpoints: Depart Date: 2020-05-28 Depart Time: 18:19:49 Available Weight: 9 KG Unwanted Categories: Cameras

Description: Who want to send something to Egypt? Send a request please.

Shipment

@ahmedomar

Date: 2020-05-28 Reward: 10.00

Description: please accept my request.

Item 1: quiz Category: Other Weight: 1 KG Quantity: 1

- Finally, when the deal is completed, the request will be moved to the Done tab, and both users will be able to review each other.

×

Details

REVIEW

Turkey ↕ Egypt

Trip

@Beyza\_Yil

Midpoints: Depart Date: 2020-05-28 Unwanted Categories: Cameras

Description: Who want to send something to Egypt? Send a request please.

Shipment

@ahmedomar

Date: 2020-05-28

Description: please accept my request.

Item 1: quiz Category: Other Weight: 1 KG Quantity: 1

Review this deal

Review


Thanks a lot. You are a nice person!

★★★★☆

CANCEL SUBMIT

- Then the review will be displayed on the user's profile, and the user's rating will be affected.

**n**delivery SEARCH LOGIN SIGN UP!


  
[Change Picture](#)  
**Ahmed Asim**  
★★★★★

ABOUT ME TRIPS SHIPMENTS REQUESTS REVIEWS

### See what other travelers and shoppers are saying about me!


1 Reviews

Thanks A Lot. You Are A Nice Person!

 **@Beyza\_Yil** ★★★★★

- Users may also display other users' profiles.

**n**delivery SEARCH MY PROFILE COMMUNITY CHAT SIGN OUT CREATE +

  
**Beyza Yillmaz**  
★★★★★

ABOUT ME TRIPS SHIPMENTS REVIEWS

**Name:** Beyza Yillmaz **Location:** Turkey

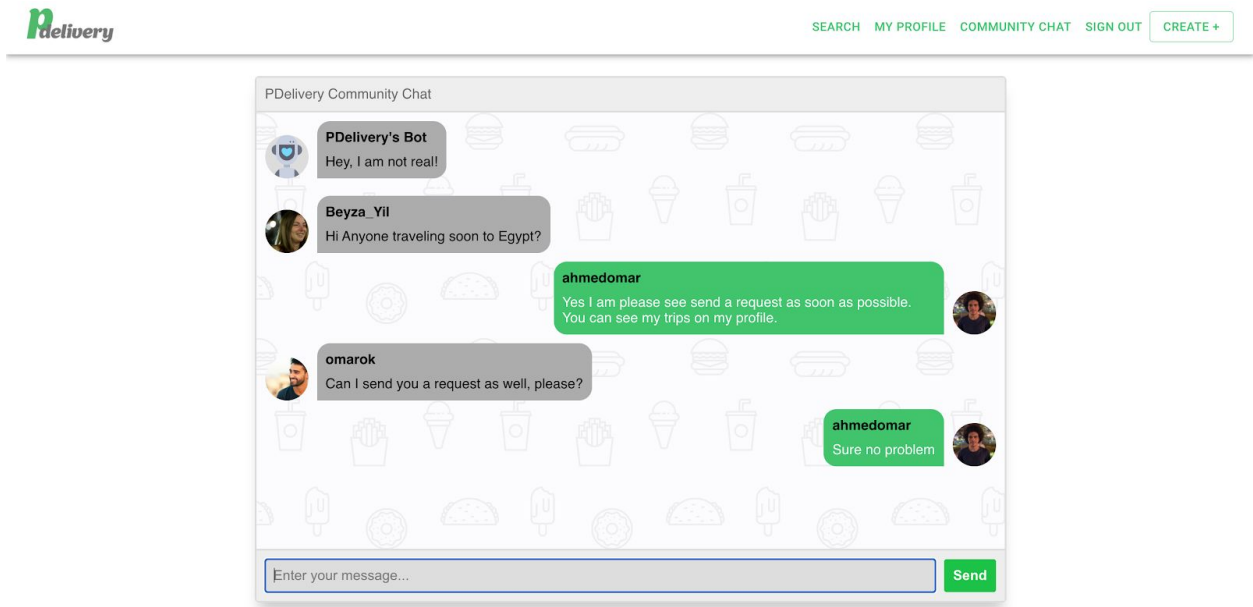
**Username:** Beyza\_Yil

**Email:** beyzayillmaz@gmail.com

UPDATE



- Finally, the users may participate in a community chat.



Thank you for reading!